Discrete Mathematics 309 (2009) 5610–5617

Contents lists available at ScienceDirect

Discrete Mathematics

journal homepage: www.elsevier.com/locate/disc

On the complexity of SNP block partitioning under the perfect phylogeny model☆

Jens Gramm [a], Tzvika Hartman [b], Till Nierhoff [c], Roded Sharan [d,*], Till Tantau [e]

[a] *Wilhelm-Schickard-Institut für Informatik, Universität Tübingen, Germany*
[b] *Department of Computer Science, Bar-Ilan University, Ramat-Gan 52900, Israel*
[c] *International Computer Science Institute, Berkeley, USA*
[d] *School of Computer Science, Tel-Aviv University, Tel-Aviv 69978, Israel*
[e] *Institut für Theoretische Informatik, Universität zu Lübeck, Germany*

ARTICLE INFO

Article history:
Received 7 January 2007
Accepted 1 April 2008
Available online 15 May 2008

Keywords:
Perfect phylogeny haplotyping
Perfect path phylogeny
Partitioning problems

ABSTRACT

Recent technologies for typing single nucleotide polymorphisms (SNPs) across a population are producing genome-wide genotype data for tens of thousands of SNP sites. The emergence of such large data sets underscores the importance of algorithms for large-scale haplotyping. Common haplotyping approaches first partition the SNPs into blocks of high linkage-disequilibrium, and then infer haplotypes for each block separately. We investigate an integrated haplotyping approach where a partition of the SNPs into a minimum number of non-contiguous subsets is sought, such that each subset can be haplotyped under the perfect phylogeny model. We show that finding an optimum partition is NP-hard even if we are guaranteed that two subsets suffice. On the positive side, we show that a variant of the problem, in which each subset is required to admit a perfect *path* phylogeny haplotyping, is solvable in polynomial time.

1. Introduction

Single nucleotide polymorphisms (SNPs) are differences in a single base across the population, within an otherwise conserved genomic sequence [24]. SNPs account for the majority of the variation between DNA sequences of different individuals [21]. Especially when they occur in coding or otherwise functional regions, variations in the allelic content of SNPs are linked to medical conditions or may affect drug response.

The sequence of alleles in contiguous SNP positions along a chromosomal region is called a *haplotype*. A SNP commonly has two variants, or *alleles*, in the population, corresponding to two of the four genomic letters A, C, G, and T. For diploid organisms, the *genotype* specifies, for every SNP position, the particular alleles that are present at this site in the two chromosomes. Genotype data contains information only on the combination of alleles at a given site; it does not reveal the association of each allele with one of the two chromosomes. Current technology, suitable for large-scale polymorphism screening, obtains only the genotype information at each SNP site. The actual haplotypes in the typed region can be obtained at a considerably higher cost [21]. Due to the importance of haplotype information in association studies, it is desirable to develop efficient methods for inferring haplotypes from genotype information.

☆ A preliminary version of this paper appeared in [Jens Gramm, Tzvika Hartman, Till Nierhoff, Roded Sharan, Till Tantau, On the complexity of SNP block partitioning under the perfect phylogeny model, in: Proc. of the sixth Workshop on Algorithms in Bioinformatics, WABI'06, 2006, pp. 92–102].

* Corresponding author.
E-mail addresses: gramm@informatik.uni-tuebingen.de (J. Gramm), hartmat@cs.biu.ac.il (T. Hartman), roded@tau.ac.il (R. Sharan), tantau@tcs.uni-luebeck.de (T. Tantau).

0012-365X/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.disc.2008.04.002

Extant approaches for inferring haplotypes from genotype data include parsimony approaches [4,15], maximum likelihood methods [9], and statistical methods [20,22]. Here we consider a perfect-phylogeny-based technique for haplotype inference, first introduced in a seminal paper by Gusfield [16]. This approach assumes that the underlying haplotypes can be arranged in a phylogenetic tree, so that for each SNP site the set of haplotypes with the same state at this site forms a connected subtree. The theoretical elegance of the perfect phylogeny approach to haplotyping as well as its efficiency and good performance in practice [3,6] have spawned several studies of the problem and its variants [1,6,18]. For more background on perfect phylogeny haplotyping see [17].

A more restricted model is the *perfect path phylogeny* model [12,13], in which the phylogenetic tree is a single long path. The motivation for considering path phylogenies is the discovery that yin-yang (complementary) haplotypes are very common in populations [25]. The presence of such haplotypes implies that under the prefect phylogeny model any phylogeny has to take the form of a path. We previously found that over 70% of publicly available human genotype matrices that admit a perfect phylogeny also admit a perfect path phylogeny [12,13]. In the presence of missing data, finding perfect path phylogenies appears to be easier since this problem is fixed-parameter tractable [13], which is not known to be the case for perfect (branching) phylogenies. This suggests that the perfect path phylogeny model is somewhat easier computationally. In this paper we give further evidence to this claim.

The perfect phylogeny assumption is particularly appropriate for short genomic regions that have not undergone recombination events. For longer regions, it is common practice to sidestep the recombination problem by inferring haplotypes only for small blocks of data and then assembling these blocks to obtain the complete haplotypes [7]. Thus, the common approach to large-scale haplotyping consists of two phases: First, one partitions the data into blocks of SNPs. Then, one infers the haplotypes for each block separately using an algorithm based on the perfect phylogeny model. Most existing block-partitioning methods partition the data into contiguous blocks, whereas in real biological data the blocks need not be contiguous [2].

In this paper we study the computational complexity of a combined approach that aims at finding a partition of an input set of SNPs into a minimum number of subsets (not necessarily contiguous), such that the genotype data induced on each subset is amenable to haplotyping under a perfect phylogeny model. We consider several variants of this problem. First, we show that for haplotype data it is possible to check in polynomial time whether there is a perfect phylogeny partition of size at most two. However, for size three and more the problem becomes NP-hard (Section 4). The situation for genotype data is even worse: Coming up with a partition into a constant number of subsets is NP-hard even if we are guaranteed that two sets suffice (Section 5). Our main result is a positive one: we show that the partitioning problem under the perfect path phylogeny model can be solved efficiently even for genotype matrices (Section 6). This result implies a novel haplotyping method that integrates the block partitioning phase and the haplotyping phase. Moreover, unlike most block-partitioning techniques, our algorithm does not assume that the blocks are contiguous.

2. Preliminaries and problem statement

In this section we provide background on haplotyping via perfect phylogeny and formulate the partitioning problems that are studied in this paper.

2.1. Haplotypes, genotypes, and perfect phylogenies

A *haplotype* is a row vector with binary entries. Each position of the vector corresponds to a SNP site, and specifies which of the two possible alleles are present at that position (we consider only bi-allelic SNPs since sites with more alleles are rare). For a haplotype h, let $h[i]$ denote the value of the ith position of h. A *haplotype matrix* is a binary matrix whose rows are haplotypes. A haplotype matrix B *admits a perfect phylogeny* or just *is pp* if there exists a rooted tree T_B such that:

(1) Every row of B labels exactly one node of T_B.
(2) Each column of B labels exactly one edge of T_B.
(3) Every edge of T_B is labelled by at least one column of B.
(4) For every two rows h_1 and h_2 of B and every column i, we have $h_1[i] \neq h_2[i]$ if and only if i lies on the path from h_1 to h_2 in T_B.

A *genotype* is a row vector with entries in $\{0, 1, 2\}$, each corresponding to a SNP site. A 0- or 1-entry in a genotype implies that the two underlying haplotypes have the same entry in this position. A 2-entry in a genotype implies that the two underlying haplotypes differ at that position. A *genotype matrix* is a matrix whose rows are genotypes. Two haplotypes h_1 and h_2 *explain* (or *resolve*) a genotype g if for each position i the following holds: $g[i] \in \{0, 1\}$ implies $h_1[i] = h_2[i] = g[i]$; and $g[i] = 2$ implies $h_1[i] \neq h_2[i]$. Given an $n \times m$ genotype matrix A and a $2n \times m$ haplotype matrix B, we say that B *explains* A if for every $i \in \{1, \ldots, n\}$ the haplotypes in rows $2i - 1$ and $2i$ of B explain the genotype in row i of A. For a genotype g and a value $v \in \{0, 1, 2\}$, the set of columns with value v in g is called the *v-set* of g. Given an $n \times m$ genotype matrix A, we say that it *admits a perfect phylogeny* or just *is pp* if there is a $2n \times m$ haplotype matrix B that explains A and admits a perfect phylogeny. The problem of determining whether a given genotype matrix admits a perfect phylogeny, and if it does, finding the explaining haplotypes, is called *perfect phylogeny haplotyping*.

Even though we use rooted trees in the definition of perfect phylogenies, the choice of the root is actually arbitrary. For the *directed* version of the perfect phylogeny problem this is no longer the case: for this version we are given the labelling

of the root as part of the input. The problem is to find explaining haplotypes for the input genotypes such that they can be arranged in a perfect phylogeny with the root labelled with the given haplotype. For this directed version, one may assume, without loss of generality, that the labelling of the root consists only of 0-entries (we can exchange the roles of 0- and 1-entries in all columns where this is not the case).

As shown in [6], one can reduce the general (undirected) problem to the directed case by using a simple transformation of the input matrix: In each column of the genotype matrix search for the first 0- or 1-entry (that is, first entry which is not a 2-entry). If this entry is a 1-entry, exchange the roles of 0-entries and 1-entries in this column.

2.2. Perfect path phylogenies

A *perfect path phylogeny* is a perfect phylogeny in the form of a path, which means that the perfect phylogeny may have at most two leaves and branching may occur only at the root. If a haplotype/genotype matrix admits a perfect path phylogeny, we say that *it is ppp.*

The motivation for considering path phylogenies in the context of haplotyping is the discovery that yin-yang (complementary) haplotypes are very common in human populations [25]. We previously found, see [13,12], that over 70% of publicly available human genotype matrices that admit a perfect phylogeny also admit a perfect path phylogeny. In the presence of missing data, finding perfect path phylogenies appears to be easier since this problem is fixed-parameter tractable, which is not known to be the case for perfect (branching) phylogenies.

2.3. Partitioning problems

Given a set C of columns of a haplotype or genotype matrix, define the following functions: $\chi_{\text{pp}}(C) = \min\{k \mid \exists C_1, \ldots, C_k : C = C_1 \cup \cdots \cup C_k,$ each C_i is pp$\}$ and $\chi_{\text{ppp}}(C) = \min\{k \mid \exists C_1, \ldots, C_k : C = C_1 \cup \cdots \cup C_k,$ each C_i is ppp$\}$. By "C_i is pp" we mean that the matrix formed by the columns in C_i is pp (the pp-property does not depend on the order of the columns). We call a partition $(C_1, \ldots, C_k)$ of C in which each C_i is pp a *pp-partition*. In a slight abuse of notation we write $\chi_{\text{pp}}(A)$ for $\chi_{\text{pp}}(C)$, when C is the set of columns in the matrix A. The notation for ppp is analogously defined.

Our objective in this paper is to determine the computational complexity of the functions χ_{pp} and χ_{ppp}, both for haplotype matrices and, more generally, for genotype matrices. The *pp-partition* problem is to compute χ_{pp} and a partition realizing the optimum value, and the *ppp-partition* problem is to compute χ_{ppp} and a corresponding partition.

Similarly to perfect phylogeny haplotyping, there are directed and undirected versions of the pp- and ppp-partition problems. However, the above-mentioned transformation of Eskin et al. [6] can be used to reduce the more general undirected case to the directed case also for the partition problems. This shows that both versions are equivalent, allowing us to restrict attention to the directed version in the following.

3. Review of related results

In this section we review results from the literature that are used in the sequel. This includes both results on haplotyping as well as results from order theory.

3.1. The complexity of perfect phylogeny haplotyping

A polynomial-time algorithm for perfect phylogeny haplotyping was first given by Gusfield [16]. A central tool in Gusfield's algorithm and those that followed it, is the concept of *induce*: The *induce* of a genotype matrix A is the set of rows that are common to all haplotype matrices B that explain A. For example, the induce of the genotype matrix $\binom{2\ \ 2\ \ 1}{1\ \ 0\ \ 0}$ is just $\{100\}$, but the induce of $\binom{0\ \ 2}{1\ \ 0}$ is $\{00, 01, 10\}$. A key theorem on perfect phylogenies is the following (cf. [14]):

Theorem 1 (*Four Gamete Test*). *A haplotype matrix B is pp if and only if the induce of any pair of its columns has size at most* 3 .

For genotype matrices, an induce of size 4 for a pair of columns also means that the matrix admits no perfect phylogeny, but the converse is no longer true and a more elaborate algorithm is needed to check whether a genotype matrix is pp.

3.2. A partial-order perspective on haplotyping

We now review previous results, mainly from [12], that relate haplotyping to order theory. As shown in [16,12], one can characterize the genotype matrices that admit a directed perfect phylogeny as follows:

Theorem 2. *A genotype matrix A admits a directed perfect phylogeny if and only if there exists a rooted tree T_A such that:*

(1) *Each column of A labels exactly one edge of T_A.*
(2) *Every edge of T_A is labelled by at least one column of A.*
(3) *For every row r of A:* (a) *the columns in its* 1*-set label a path from the root to some node u; and* (b) *the columns in the* 2*-set of row r label a path that visits u and is contained in the subtree rooted at u.*

We consider the following partial orders on the columns of A:

(1) The *ancestor relation* induced by the column labels of T_A.
(2) The *partial order* $\succeq$: Let $1 \succ 2 \succ 0$ and extend this order to $\{0, 1, 2\}$-columns by setting $c \succeq c'$ if $c[i] \succeq c'[i]$ for all rows i.
(3) The *leaf count order:* The *leaf count* of a column c is twice the number of 1-entries plus the number of 2-entries in c. This relation orders columns by increasing leaf count and considers columns as not comparable if they are different but have the same leaf count.

The first and the last order were introduced by Gusfield [16]; the order $\succeq$ was introduced by Eskin et al. [6], who implicitly showed that each order extends the one above it. Note that the last two relations exist even when there is no perfect phylogeny for A. In particular, they can be computed before the tree T_A is known.

The following theorem shows that the existence of a perfect path phylogeny for a matrix A with column set C can be decided based on the properties of $(C, \succeq)$ alone, but we first need a definition.

Definition 1. Two columns are *separable* if each has a 0-entry in the rows where the other has a 1-entry. We say that a set C of $\{0, 1, 2\}$-columns has the *ppp-property* if it can be covered by two (possibly empty) chains $(C_1, \succeq)$ and $(C_2, \succeq)$, so that their maximal elements are separable (if both are non-empty). The pair (C_1, C_2) is called a *ppp-cover* of C.

Theorem 3 ([12]). *A genotype matrix A admits a directed perfect path phylogeny if and only if its column set has the ppp-property.*

3.3. Colourings of hypergraphs

A *hypergraph* $H = (V, E)$ consists of a *vertex set* V and a *set* E *of hyperedges*, which are subsets of V. A hypergraph is *k-uniform* if each edge has exactly k elements. A *legal χ-colouring* of a hypergraph H is a function $f: V \to \{1, \ldots, \chi\}$ such that no edge in E is monochromatic. The *chromatic number of H* is the minimum χ for which there exists a legal χ-colouring of H.

It is folklore that one can check in polynomial time whether a graph (a 2-uniform hypergraph) can be 2-coloured, and that checking whether it can be χ-coloured is NP-hard for every $\chi \geq 3$ [11]. This implies that, for every $k \geq 2$ and every $\chi \geq 3$, checking whether a k-uniform hypergraph is χ-colourable is NP-hard. It is even NP-hard to approximate the chromatic number within a factor of n^ϵ, see [19].[1]

4. PP-partitioning problems for haplotype matrices

In this section we study the complexity of $\chi_{\text{pp}}(B)$ for *haplotype* matrices B. It turns out we can decide in polynomial time whether $\chi_{\text{pp}}(B)$ is 1 or 2, but it is NP-hard to decide whether it is 3 or more. The proofs of these results rely on reductions from χ_{pp}, restricted to haplotype matrices, to graph colouring and back. The hardness proof does not carry over to perfect *path* phylogenies. Indeed, we will see later that χ_{ppp} is polynomial-time computable *even for genotype matrices*.

Theorem 4. *There is a polynomial-time algorithm that checks, on input of a haplotype matrix B, whether $\chi_{\text{pp}}(B) \leq 2$.*

Proof. By Theorem 1 we can check in polynomial time whether $\chi_{\text{pp}}(B) = 1$. To check whether $\chi_{\text{pp}}(B) \leq 2$, we construct the following graph on the columns of the matrix B: We add an (undirected) edge between every two columns whose induce has size 4. We claim that $\chi_{\text{pp}}(B) \leq 2$ if and only if the resulting graph can be coloured with two colours. To see this, note that if the chromatic number of the graph is larger than 2, then any subset of the columns of B will contain two columns having an induce of size 4. On the other hand, if the graph is 2-colourable, then the column sets corresponding to the two colour classes constitute a covering of the matrix B. Furthermore, by definition, none of the sets contains two columns having an induce of size 4. Hence, by Theorem 1, each of the column sets is pp. ■

Theorem 5. *For every $k \geq 3$, it is* NP*-hard to pp-partition a haplotype matrix B into k perfect phylogenies.*

Proof. We prove the claim by presenting a reduction from the NP-hard k-COLOURING problem [11] to pp-partitioning a haplotype matrix into k perfect phylogenies.

Reduction. Let $G = (V, E)$ be an input graph for k-COLOURING. We map it to the following haplotype matrix B: There is a column for each vertex $v \in V$. The first row in B is an all-0 row. For each vertex v there is one row having a 1 in column v and having 0's in all other columns. Finally, for each edge $\{u, v\} \in E$ there is a row in B having 1-entries in columns u and v and having 0-entries in all other columns.

Correctness. Consider a colouring of the graph G. This colouring induces a partition of the columns of the matrix B. For any two columns in the same class of the partition, the induce will not contain $\{11\}$ and, thus, this class admits a perfect phylogeny

[1] Strictly speaking the approximation problem itself is not a language and, thus, cannot be "NP-hard." By "it is NP-hard to approximate the chromatic number with a factor of n^ϵ" we mean that all problems in NP can be many-one reduced to the chromatic number problem in such a way that for all graphs G output by the reduction we either have $\chi(G) \leq \alpha$ or $\chi(G) \geq \alpha n^\epsilon$ for some α.

by Theorem 1. For the other direction, consider a partition of B into perfect phylogenies. Inside each class the induce of any two different columns must have size at most 3. Since the induce of any two different columns always contains 00, 01, and 10, the induce must be missing 11. Hence, for any two columns in the same class there cannot be an edge in G. Thus, the partition induces a colouring of the graph G. ■

The theorem also implies hardness of approximation for the problem:

Theorem 6. *Unless* P = NP, *the function* χ_{pp} *cannot be approximated within a factor of* n^{ϵ} *for any* $\epsilon > 0$.

Proof. In the reduction given in the proof of Theorem 5 the number of perfect phylogenies directly corresponds to the number of colours in a colouring. The colouring problem for graphs is NP-hard to approximate to within a factor of n^{ϵ}, see [19]. ■

5. PP-partitioning problems for genotype matrices

By the results of the previous section there is little hope of finding (or even coming close to) the minimum number of perfect phylogenies that cover a haplotype matrix. Since haplotype matrices are just restricted genotype matrices (namely, genotype matrices with no 2-entries), the situation for genotype matrices can even be worse. Indeed, we show that for genotype matrices even if two perfect phylogenies suffice, coming up with a partition into any constant number χ of perfect phylogenies is still NP-hard.[2]

Theorem 7. *For every* $\chi \geq 2$, *it is* NP*-hard to come up with a pp-partition of a genotype matrix* A *into* χ *classes, even if we know that* $\chi_{\text{pp}}(A) \leq 2$ *holds.*

Proof. We reduce the problem of colouring a 3-uniform, 2-colourable hypergraph with a constant number of colours to the pp-partition problem; the former problem, is known to be NP-hard, see [5].

Reduction. Given a 3-uniform hypergraph H, construct A as follows: A has four rows per hyperedge and one column per vertex. For each hyperedge $h = \{u, v, w\}$, the submatrix of A corresponding to the rows for h and to the columns for u, v, and w is the matrix $S := \begin{pmatrix} 2 & 2 & 2 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$. Every entry of A not contained in such a submatrix is 0.

Correctness. We show how to construct a pp-partition of the columns of A into k sets given a k-colouring of H, and how to construct a k-colouring of H given a pp-partition into k sets.

Given a k-colouring of H with colour classes $V_1, \ldots, V_k$, let C_i be the columns corresponding to the vertices of V_i. We claim that each C_i is pp. To this end, let A_i denote the submatrix of A that consists of the columns C_i. Each row contains either one 1-entry or up to two 2-entries and otherwise the rows contain only 0-entries: No row can contain three or more 2-entries, because the maximum number of 2-entries per row of A is three and the columns of these entries cannot all be contained in C_i, since V_i does not contain whole hyperedges.

Those rows that do not contain any 2-entries are resolved trivially by having two copies of these rows in the haplotype matrix. Those containing 2-entries are replaced by two haplotype rows as follows: If they contain at most one 2-entry, they are replaced by two copies in which the 2-entry is substituted by a 0- and a 1-entry. If they contain two 2-entries, in the first copy the 2-entries are replaced by a 0- and a 1-entry (in this order), in the second copy they are replaced by a 1- and a 0-entry (in this order). Other than 2-entries, these rows only contain 0-entries; so the haplotypes they are replaced by have only one 1-entry.

This way of resolving the genotypes in A_i into haplotypes leaves at most one 1-entry per row, which implies that the haplotype matrices are pp by the four-gamete test (Theorem 1).

Given a pp-partition $(C_1, \ldots, C_k)$ of the columns of A, let V_i contain the vertices corresponding to the set C_i. We claim that no V_i contains a complete hyperedge in H. Assume for a contradiction that $u, v, w \in C_i$ for some i and that $h = \{u, v, w\}$ is an edge in H. Then, by the reduction, the submatrix A_i, consisting of the columns C_i, contains the submatrix S. Consider a replacement of the first row with a consistent haplotype pair. One of the haplotypes has to contain two 1-entries and, consequently, there is a pair of columns that induces all four gametes, a contradiction. ■

6. A polynomial-time algorithm for ppp-partitioning genotype matrices

In this section we give a polynomial-time algorithm for ppp-partitioning of genotype matrices. The algorithm is based on reducing this problem to the problem of finding a maximum matching in a graph, which can be solved in polynomial time [8].

[2] Once more the partitioning problem itself cannot be "NP-hard." The exact claim is that every problem in NP can be reduced to the pp-partitioning problem in such a way that for all genotype matrices A output by the reduction, either $\chi_{\text{pp}}(A) \leq 2$ or $\chi_{\text{pp}}(A) > \chi$.

algorithm PPP-PARTITIONING
  let $G \leftarrow (C^{\text{left}} \cup C^{\text{right}}, E_{\text{green}} \cup E_{\text{red}})$
  let $M \leftarrow \text{maximum\_matching}(G)$
  let $G \leftarrow (C^{\text{left}} \cup C^{\text{right}}, M)$
  let $G' \leftarrow G$ with each pair $\{c^{\text{left}}, c^{\text{right}}\}$ contracted to a single vertex
  foreach connected component X of G' **do**
    output the perfect path phylogeny corresponding to X

Fig. 1. A polynomial-time algorithm for finding a ppp-partition.

Let A be a genotype matrix and let C be the set of columns of A. We form two copies of C, called the *left* and the *right copy*, by adding appropriate superscripts to the elements of C: Let $C^{\text{left}} := \{c^{\text{left}} \mid c \in C\}$ and $C^{\text{right}} := \{c^{\text{right}} \mid c \in C\}$. These two copies can be envisioned as being drawn on the left and right side of a page. We define a set $E_{\text{green}} := \{(c^{\text{left}}, d^{\text{right}}) \mid c \succ d\}$ of green edges, which interconnects the two copies, and a set $E_{\text{red}} := \{(c^{\text{right}}, d^{\text{right}}) \mid c \text{ and } d \text{ are separable}\}$, which connects only vertices in the right copy. Fulkerson's reduction of Dilworth's Theorem to the König–Egerváry Theorem consists mainly of the observation that each matching M in the bipartite graph $(C^{\text{left}}, C^{\text{right}}, E_{\text{green}})$ corresponds one-to-one to a partition of $(C, \succeq)$ into $|C| - |M|$ chains (see [10] for more details). Our method for computing $\chi_{\text{ppp}}(A)$ relies on the following modification of that observation, where the set of red edges is also taken into account (note that this transforms the graph from a bipartite graph into a general graph):

Lemma 8. *Let k be a number. Then there exists a matching M of the graph $G = (C^{\text{left}} \cup C^{\text{right}}, E_{\text{green}} \cup E_{\text{red}})$ of size $|M| = |C| - k$ if and only if there exists a partition of the set of columns C into $k = |C| - |\mathcal{M}|$ subsets such that each subset admits a directed perfect path phylogeny.*

In other words, the matchings of G are in one-to-one correspondence with the different partitions of C into directed perfect path phylogenies.

Proof. Let M be a matching of G of size $|M| = |C| - k$. Convert M into a set M' by forgetting about the side of the vertices, that is, $M' = \{\{u, v\} \mid \{u^{s_1}, v^{s_2}\} \in M, s_1, s_2 \in \{\text{left}, \text{right}\}\}$. Note that each edge in M' inherits exactly one colour from the corresponding edge in M (it cannot inherit two colours since this would imply that a vertex in the right copy is matched twice). Let us make some observations about the resulting graph $G' = (C, M')$.

First, G' has maximum degree 2: Consider a vertex $c \in C$ in the graph G'. Then c^{left} and c^{right} are both connected to at most one vertex in G since M is a matching. Thus, in G' the vertex c is connected to at most two vertices.

Second, we claim that for every path $(c_1, c_2, \ldots, c_n)$ in G' that uses only green edges we either have $c_1 \succ c_2 \succ \cdots \succ c_n$ or $c_1 \prec c_2 \prec \cdots \prec c_n$: Since $\{c_1, c_2\}$ is a green edge in G', we must have $\{c_1^{\text{left}}, c_2^{\text{right}}\} \in E_{\text{green}}$ or $\{c_1^{\text{right}}, c_2^{\text{left}}\} \in E_{\text{green}}$. By definition, the first case implies $c_1 \succ c_2$. Consider the green edge $\{c_2, c_3\}$ in G'. Since c_2^{right} is already matched by M, we can conclude that $\{c_2^{\text{left}}, c_3^{\text{right}}\} \in E_{\text{green}}$ and, thus, $c_2 \succ c_3$. Using the same argument repeatedly, we can inductively conclude $c_1 \succ c_2 \succ \cdots \succ c_n$ as claimed. For the second case, where $\{c_1^{\text{right}}, c_2^{\text{left}}\} \in E_{\text{green}}$, the definition yields $c_1 \prec c_2$ and a similar argument as before shows $c_1 \prec c_2 \prec \cdots \prec c_n$.

Third, we claim that every connected component of G' has one of two possible forms:

(1) It is a path $(c_1, c_2, \ldots, c_n)$ connected by green edges, such that $c_1 \succ c_2 \succ \cdots \succ c_n$.
(2) It contains a red edge $\{c_1, d_1\}$ and the remaining vertices in the component form two disjoint paths $(c_1, c_2, \ldots, c_n)$ and $(d_1, \ldots, d_m)$ of green edges, such that $c_1 \succ c_2 \succ \cdots \succ c_n$ and $d_1 \succ d_2 \succ \cdots \succ d_m$.

To prove this, first assume that the component contains no red edges. We know already that G' has maximum degree 2, so the component must have the form of a path or a cycle. But, we saw already that $c_1 \succ c_2 \succ \cdots \succ c_n$ if we name the vertices in the correct order, which implies in particular that the component is not a cycle. Now assume that the component contains a red edge $\{c_1, d_1\}$. Then $\{c_1^{\text{right}}, d_1^{\text{right}}\} \in E_{\text{red}}$. Let $(c_1, c_2, \ldots, c_n)$ be the path in G' leading away from c_1 up to either the end of the path or up to a red edge. Likewise, let $(d_1, \ldots, d_m)$ be the green path leading away from d_1 up to the end or up to a red edge. Because c_1^{right} is already matched in M (via the red edge), as before we can conclude that $c_1 \succ c_2 \succ \cdots \succ c_n$. Likewise, we can conclude $d_1 \succ d_2 \succ \cdots \succ d_m$. Finally, we know that $\{c_{n-1}^{\text{left}}, c_n^{\text{right}}\} \in M$ and $\{d_{m-1}^{\text{left}}, d_m^{\text{right}}\} \in M$. This means that c_n and d_m cannot be connected by a red edge and they also cannot be identical.

We are now ready to claim that each vertex set of a component of (C, M') has the ppp-property: As we just saw, each component of G' induces either a chain in $(C, \succeq)$ or it induces two chains whose top elements are separable. By Theorem 3, this means that corresponding sets of columns admit a directed perfect path phylogeny. Furthermore, the earlier argument shows that G' must be acyclic. This implies that the number of connected components of $G' = (C, M')$ is exactly $|C| - |M'| = |C| - |M| = k$.

For the second direction, let $C_1, \ldots, C_k$ be a partition of C into subsets that have the ppp-property. Each C_i gives rise to a matching of size $|C_i| - 1$ in the induced subgraph $G[C_i^{\text{left}} \cup C_i^{\text{right}}]$. The union of these matchings is disjoint and, therefore, a matching of size $|C| - k$. ■

We now arrive at our main result:

Theorem 9. *There exists a polynomial-time algorithm for the ppp-partitioning problem, which runs in time* $O((n + \sqrt{m})m^2)$*, where* n *is the number of genotypes and* m *is the number of SNP sites.*

Proof. The ppp-partitioning algorithm is summarized in Fig. 1. The correctness of the algorithm is implied by Lemma 8. As for the running time, we first note that the graph G has $O(m)$ vertices and $O(m^2)$ edges. Checking the existence of each edge is done in time $O(n)$ and, thus, G is constructed in time $O(nm^2)$. Finding a maximum matching requires time $O(m^{2.5})$ (see [8]), and partitioning into connected components can be easily done within this time bound. Hence, the total running time of the algorithm is $O(m^2(n + \sqrt{m}))$. ■

Notice that typically there is more than one maximum matching and hence more than one optimal ppp-partition. The enumeration of all optimal solutions can be done in time $O(m)$ per solution [23].

7. Concluding remarks

In this paper we studied the complexity of SNP block partitioning under the perfect phylogeny model. We showed that although the partitioning problems are NP-hard for the perfect phylogeny model, they are tractable for the more restricted perfect path phylogeny model. The contribution is two-fold. On the theoretical side, this demonstrates again the power of the perfect path phylogeny model. On the practical side, we present a block partitioning protocol that integrates the block partitioning phase and the haplotyping phase. We note, however, that there may be an exponential number of minimal partitions and, thus, in order to choose the most biologically meaningful solution we might need to consider also some other criteria for block partitioning. Future directions may include testing the algorithm on real data, and comparing this method with other block partitioning methods. Also, it would be interesting to explore the space of optimal solutions in order to find the most relevant one.

Acknowledgments

JG was supported by a grant for the DFG project *Optimal solutions for hard problems in computational biology*. JG, TN and TT were supported through a postdoc fellowship by the DAAD. TT was supported by a grant for the DFG project *Complexity of haplotyping problems.* RS was supported by an Alon Fellowship.

References

[1] V. Bafna, D. Gusfield, G. Lancia, S. Yooseph, Haplotyping as perfect phylogeny: A direct approach, Journal of Computational Biology 10 (3–4) (2003) 323–340.
[2] C.S. Carlson, M.A. Eberle, L. Kruglyak, D.A. Nickerson, Mapping complex disease loci in whole-genome association studies, Nature 429 (2004) 446–452.
[3] R.H. Chung, D. Gusfield, Empirical exploration of perfect phylogeny haplotyping and haplotypers, in: Proceedings of the 9th International Conference on Computing and Combinatorics, in: LNCS, vol. 2697, Springer, 2003, pp. 5–19.
[4] A.G. Clark, Inference of haplotypes from PCR-amplified samples of diploid populations, Journal of Molecular Biology and Evolution 7 (2) (1990) 111–122.
[5] I. Dinur, O. Regev, C.D. Smyth, The hardness of 3-uniform hypergraph coloring, Combinatorica 25 (2005) 519–535.
[6] E. Eskin, E. Halperin, R.M. Karp, Efficient reconstruction of haplotype structure via perfect phylogeny, Journal of Bioinformatics and Computational Biology 1 (1) (2003) 1–20.
[7] E. Eskin, E. Halperin, R. Sharan, A note on phasing long genomic regions using local haplotype predictions, Journal of Bioinformatics and Computational Biology 4 (2006) 639–647.
[8] S. Even, O. Kariv, An $o(n^{2.5})$ algorithm for maximum matching in general graphs, in: Proceedings of the 16th Symposium on Foundations of Computer Science, FOCS, 1975, pp. 100–112.
[9] L. Excoffier, M. Slatkin, Maximum-likelihood estimation of molecular haplotype frequencies in a diploid population, Molecular Biology and Evolution 12 (5) (1995) 921–927.
[10] S. Felsner, V. Raghavan, J. Spinrad, Recognition algorithms for orders of small width and graphs of small Dilworth number, Order 20 (2003) 351–364.
[11] M. Garey, D. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, San Francisco, 1979.
[12] J. Gramm, T. Nierhoff, R. Sharan, T. Tantau, Haplotyping with missing data via perfect path phylogenies, Discrete Applied Mathematics 155 (2007) 788–805.
[13] J. Gramm, T. Nierhoff, T. Tantau, Perfect path phylogeny haplotyping with missing data is fixed-parameter tractable, in: Proceedings of the 2004 International Workshop on Parameterized and Exact Computation, in: Lecture Notes in Computer Science, vol. 3162, Springer-Verlag, 2004, pp. 174–186.
[14] D. Gusfield, Efficient algorithms for inferring evolutionary trees, Networks 21 (1991) 19–28.
[15] D. Gusfield, Inference of haplotypes from samples of diploid populations: Complexity and algorithms, Journal of Computational Biology 8 (3) (2001) 305–323.
[16] D. Gusfield, Haplotyping as perfect phylogeny: Conceptual framework and efficient solutions, in: Proceedings of the Sixth Annual International Conference on Computational Molecular Biology, RECOMB, ACM Press, 2002, pp. 166–175.
[17] Dan Gusfield, Steven Hecht Orzack, Combinatorial methods for haplotype inference, in: Handbook of Computational Molecular Biology, CRC Press, 2005.
[18] E. Halperin, R.M. Karp, Perfect phylogeny and haplotype assignment, in: Proceedings of the Eighth Annual International Conference on Computational Molecular Biology, RECOMB, ACM Press, 2004, pp. 10–19.
[19] C. Lund, M. Yannakakis, On the hardness of approximating minimization problems, Journal of the ACM 45 (5) (1994) 960–981.
[20] T. Niu, S. Qin, X. Xu, J. Liu, Bayesian haplotype inference for multiple linked single nucleotide polymorphisms, American Journal of Human Genetics 70 (1) (2002) 157–169.

[21] N. Patil, A.J. Berno, D.A. Hinds, et al., Blocks of limited haplotype diversity revealed by high-resolution scanning of human chromosome 21, Science 294 (5547) (2001) 1719–1723.
[22] M. Stephens, N. Smith, P. Donnelly, A new statistical method for haplotype reconstruction from population data, American Journal of Human Genetics 68 (4) (2001) 978–989.
[23] T. Uno, Algorithms for enumerating all prefect, maximum and maximal matchings in bipartite graphs, in: Proceedings of the International Symposium on Algorithms and Computation, 1997, pp. 92–101.
[24] D.G. Wang, J.B. Fan, C.J. Siao, A. Berno, P.P. Young, et al., Large-scale identification, mapping, and genotyping of single nucleotide polymorphisms in the human genome, Science 280 (5366) (1998) 1077–1082.
[25] J. Zhang, W.L. Rowe, A.G. Clark, K.H. Buetow, Genomewide distribution of high-frequency, completely mismatching SNP haplotype pairs observed to be common across human populations, American Journal of Human Genetics 73 (5) (2003) 1073–1081.